Goldschlag 112305CON

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent Application

Inventor(s)	David M. Goldschlag et al.		
Case	112305CON	**Conf. No.**	1089
Serial No.	09/635,778	**Group Art Unit**	3622
Filing Date	August 11, 2000		
Examiner	John W. Van Bramer		
Title	**System and Method for Voting**		

COMMISSIONER FOR PATENTS
ALEXANDRIA, VA 22313-1450

SIR:

DECLARATION UNDER 37 CFR §1.131

In association with the filing a reply to a Final Office action dated March 31, 2006, applicants are submitting for acceptance by the Examiner this Declaration under Rule 131 to antedate the Challener et al. reference cited by the Examiner in the 35 USC § 102(e) rejections.

Indeed, in accordance with Rule 131, applicants declare the following to provide a showing of "conception of the claimed invention prior to the effective reference date, and due diligence at least from immediately prior to the effective date of the reference until a subsequent reduction practice or the filing of a patent application". In particular, applicants declare the following:

Attachment B is US Patent

- The **effective date** of the Challener et al. reference, US Patent 6,081,793, is the US filing date of **December 30, 1997;**
- Applicants' conception of the claimed invention occurred **prior to February 24, 1997,** as evidenced by Attachment A, an email (dated 4 March 1997) and attachment from inventor Stuart Stubblebine to the AT&T patent division requesting a patent to be filed on the attachment that was "presented at the Financial Cryptography Conference on 24 Feb. 1997";
- Applicants applied **due diligence** toward a reduction to practice of the present invention (**at least immediately prior to the effective date of December 30, 1997**), as evidenced by Attachment B, the front page of US Patent 6,108,644, which is the parent application to the present application, denoting the filing of parent application (Serial No. 09/025,802) on **February 19, 1998**; and
- Applicants provided a **constructive reduction to practice** by the filing of the subject application on **August 11, 2000**, as a continuation of the above-noted parent application, as evidenced by Attachment C, a copy of the filing receipt for this application.

Based on the above declarations, applicants thus assert prior conception and diligent reduction to practice of the present invention sufficient to antedate the Challener et al. reference. Applicants thus request the Examiner to remove this reference as applicable to this application.

Respectfully submitted,

David M. Goldschlag et al.

By: Wendy W. Koba

Wendy W. Koba
Reg. No. 30509
Attorney for applicants
610-346-7112

Date: 6/6/06

Attachment A

Subject: new patent work
Date: Tue, 4 Mar 1997 17:24:21 +0500
From: "Stuart Stubblebine" <stubblebine@research.att.com>
Reply-To: stubblebine@research.att.com
To: samuelh@mail.att.net
CC: mar@research.att.com, dpm@research.att.com

I'm following up on a recent discussion we had on patenting some work I did with two government employees at the Navel Research Lab. It appears now that it would be beneficial for this work to be submitted through the AT&T patent process. The work was presented at the Financial Cryptography Conference on 24 Feb. 1997. The paper is titled " Serial Unlinkable Transactions". This work was of considerable interest to an employee at Encyclopedia Britannica. They are interested in prototyping the technology for possible integration into their on-line service offering. (My co-authors/inventors plan to follow up with him on this activity.)

Our secretary, Marion Riley, is sending a copy of the papers and slides.

Thanks,
Stuart Stubblebine

Stuart Stubblebine
AT&T Research, 2A-345A
600 Mountain Ave., Murray Hill, NJ 07974
stubblebine@research.att.com
voice: 908-582-5481
fax: 908-582-5192

 02/06/97 08:22:58

Unlinkable Serial Transactions
(FC97 Preproceedings Draft)

Paul F. Syverson
syverson@itd.nrl.navy.mil

Stuart G. Stubblebine
stubblebine@research.att.com

David M. Goldschlag
goldschlag@itd.nrl.navy.mil

Abstract

We present a protocol for unlinkable serial transactions suitable for a variety of network-based subscription services. The protocol prevents the service from tracking the behavior of its customers while protecting the service vendor from abuse due to simultaneous or "cloned" usage from a single subscription. We present variants of the protocol supporting pay-per-use transactions within a subscription. We describe other applications including contracting out subscription management, multivendor package sales, proof of group membership, and voter registration.

1 Introduction

This paper is motivated by an apparent conflict of interest concerning the privacy of information in an electronic exchange. Commercial service providers would like to be sure that they are paid for their services and protected from abuse due to simultaneous or "cloned" usage from a single subscription. To this end they have an interest in keeping a close eye on customer behavior. On the other hand customers have an interest in the privacy of their personal information, in particular the privacy of profiles of their commercial activity. One well known approach to this problem is to allow a customer to register with vendors under pseudonyms, one for each vendor [3]. By conducting transactions using anonymous electronic cash (e-cash) the customer's anonymity is maintained. But, the vendor is able to protect his interests by maintaining a profile on each of his anonymous customers.

In this paper we present effectively the opposite solution to this problem. The customer may be known to the vendor, but his behavior is untraceable. This would appear infeasible. If transactions cannot be linked to the customer, what is to keep him from abusing the service? For example, if someone fails to return a rented video, the video rental company would like at minimum to be sure that this person cannot rent any more videos. But, the company cannot do this if they cannot determine who the renter is.[1] We will present a protocol that makes transactions unlinkable but protects vendors from such abuses.

For the near future at least, a large part of the market on the Internet and in other electronic venues will rely on credit card based models (such as SET [16] or Cybercash [6] or simply sending credit card numbers over SSL). Applications of our protocol that require payment are not dependent on the payment mechanisms used. Thus, our protocol can be easily applied now but is equally amenable to use with e-cash. Even in an environment in which pseudonyms and anonymous e-cash are generally available vendor profiles of customers

[1] In a pseudonym based scheme, such a customer could try to open an account under a new pseudonym, but there are mechanisms to make this difficult [4]. Thus, the interests of the vendor can be protected.

(or their pseudonyms) might be undesirable because the customer's anonymity protection has a single point of failure. If the vendor is ever able to link a pseudonym to a customer, the entire profile immediately becomes linked to that customer. In our solution, if a customer is ever linked to a transaction, only his link to the one transaction is revealed. (This is somewhat analogous to the property of perfect forward secrecy in key establishment protocols.)

On what applications could our approach be used? Consider a subscription service for an on-line newspaper or encyclopedia. Customers might have an interest in keeping the searches they conduct private. At the same time, vendors would like to make it difficult for customers to transfer their ability to access the service. This will serve as our primary example.

We will also consider other applications. One example is pay-per-use service within a subscription (e.g., Lexis-Nexis or pay-per-view movies available to cable TV subscribers). Unlinkable serial transactions can also be used to provide multivendor packages as well as ongoing discounts. And, they can be used for anonymous proof of membership for applications having nothing directly to do with electronic commerce. Applications include proof of age and proof of residency. They can also be used to construct a simple voter registration protocol.

The paper is organized as follows. In section 2 we describe related work. Most of the basic mechanisms on which we rely come from work on e-cash; although, we are able to greatly simplify some of those mechanisms for our purposes. We describe these and their relation to our work. We also rely on the assumption that communicating parties will not be identified by the communications medium, independent of the messages they send. Services that prevent this are discussed as well. In section 3 we will describe the basic protocol including set up, usage, and termination of a subscription. We also discuss recovery from broken connections. In section 4 we describe various applications of unlinkable serial transactions and associatied protocol variants. In section 5 we present concluding remarks.

2 Related Work

2.1 Digital Cash

Digital cash, especially anonymous e-cash as presented by Chaum et al. [5], is characterized by several requirements [12]: independent of physical requirements, unforgeable and uncopyable, untraceable purchases, off-line, transferable, and subdividable. No known e-cash system has all of these properties, and certain properties, especially e-cash that can be divided into unlinkable change, tend to be computationally expensive.

E-cash can either be on-line or off-line In an on-line scheme, the vendor can verify with a bank that the cash has not previously been used before completing the transaction. In an off-line scheme, double spending must be detectable later, and the identity of the double spender must then be revealed. Previously agreed upon penalties can then be applied that make double spending not cost effective.

Chaum's notion of *blinding* [4] is a fundamental technique used in anonymous e-cash and assigning pseudonyms. A bank customer may want a certain amount of e-cash from the bank, but may not trust the bank not to mark (and record) the e-cash in some way. One solution is for the bank to sign something for the customer that the bank cannot read, while the customer presents the bank with evidence that the bank is signing something legitimate.

Chaum's blinding depends on the commutativity of modular multiplication operations. Therefore, the customer can create an e-cash certificate, multiply it by a random number called a blinding factor. If the

bank signs the blinded certificate, the customer can then divide out the blinding factor. The result is the unblinded certificate signed by the bank. But the bank does not know what it signed.

How can the customer assure the bank that the blinded certificate is legitimate? In Chaum's scheme, the customer presents the bank with many blinded certificates that differ in serial number, perhaps, but not in denomination. The bank chooses the one it will sign and asks the customer for the blinding factors of the others. If the randomly chosen certificates turn out to be legitimate when unblinded, the bank can have confidence that the remaining blinded certificate is legitimate too.

One on-line e-cash scheme is presented in [15]. To obtain an e-cash certificate that only he can use, a customer presents the bank with a hash of a random number. The bank signs an e-cash certificate linking that hash with a denomination. To use the e-cash, the customer reveals the random number to a vendor, who in turn takes the e-cash to a bank. Since hashes are one-way functions, it would be very hard for someone other than the customer to guess the secret that allows the e-cash to be spent. After the money is spent, the bank must record the hash to prevent it from being spent again. This scheme can be combined with blinding, to hide the actual e-cash certificate from the bank during withdrawal.

One off-line e-cash scheme is presented in [9]. There, the bank signs blinded certificates. To spend the e-cash, the customer must respond to a vendor's challenge. The response can be checked by inspecting the e-cash. Double spending is prevented because the challenge/response scheme is constructed so the combination of responses to two different challenges reveals the identity of the customer. As long as the customer does not double spend, his identity is protected. Nobody but the customer can generate responses, so the customer cannot be framed for double spending.

It may be the case that truly anonymous unlinkable e-cash enables criminal activity. Several key escrow or trustee-based systems [2] have been developed that can reveal identities to authorities who obtain proper authorizations.

Our notion of unlinkable certificates came from asking the following question: what else shares some of the features of digital cash? Unlinkable certificates share many of these features: they must preserve the user's anonymity and not be traceable, but they must protect the issuer and not be forgeable or copyable. Unlike e-cash, however, transferability is not desirable. We use hashing of random numbers and blinding in our development of unlinkable certificates. Our unlinkable certificates differ from Chaum's pseudonyms [4] which are an alternative to a universal identification system. Each pseudonym is supposed to identify its owner to some institution and not be linkable across different institutions. Unlinkable serial certificates are designed to be unlinkable both across institutions and across transactions within a single institution. In particular, we want the vendor to be unable to link transactions to a single customer, even if that customer had to identify himself initially (i.e., during the subscription process). At the same time, the vendor needs to be able to protect himself against customers that abuse his service.

Our blinding also differs from the usual approach. Typically some mechanism is necessary to assure either the issuing bank or receiving vendor that the certificate blindly signed by the issuer has the right form, i.e., that the customer has not tricked the signer into signing something inappropriate. We described Chaum's basic approach to doing this above. By moving relevant assurances to other parts of the protocols, we are able to eliminate the need for such verification. The result is a great simplification of the blinding scheme.

2.2 Anonymity Services

How can a customer keep his private information private if communication channels reveal identities? For example, vendors having toll-free numbers can subscribe to services that reveal callers' phone numbers to the vendor thereby obviating any pseudonym the customer may be using. A similar service in the form of

caller-id is now available to many private customers. If a communication channel implicitly reveals identities, how can customer's private information be protected?

The solution lies in separating identification from connections. The connection should not reveal information. Identifying information should be carried over the connection. (Of course, vendors and private parties are welcome to close connections that do not immediately provide sufficient identifying information.) On the Internet, depending upon ones environment and threat model, several solutions exist.

For e-mail, anonymous remailers can be used to forward mail through a service that promises not to reveal the sender's identity to the recipient. User's worried about traffic analysis can use Babel [11] or other Mixmaster [7] based remailers which forward messages through a series of Chaum mixes [3]. Each mix can identify only the previous and next mix, and never (both) the sender and recipient.

For Web browsing, the Anonymizer [1] provides a degree of protection. Web connections made through the Anonymizer are anonymized. By looking at connection information, packet headers,etc. the destination Web server can only identify that the connection came from (through) the Anonymizer.

Onion routing [17, 14, 10] provides anonymizing services for a variety of Internet services over connections that are resistant to traffic analysis. Like Babel, onion routing can be used for e-mail. Onion routing can also be used to hide Web browsing, remote logins, and file transfers. If the communicating parties have secure connections to endpoint onion routers, communication can be anonymous to both the network and observers, but the parties may reveal identifying information to each other. The goal of onion routing is anonymous connections, not anonymous communication. Other application independent systems that complicate traffic analysis in networks have been designed or proposed. In [8] a cryptographically layered structure similar to onions in onion routing is used to forward individual IP packets through a network, essentially building a connection for each packet in a connectionless service. In [13], mixes are used to make an ISDN system that hides the individual within a local switch originating or receiving a call.

3 Transaction Unlinkability

In this section we describe protocols that prevent linking of a client's transactions to each other. Consequently, they also cannot be linked to the client himself. We assume that the client has subscribed to a service with whom he will conduct these transactions and has provided adequate identifying and billing information (e.g., credit card numbers). The protocols make use of many basic e-cash primitives but are generally much simpler than protocols using these primitives in their more common applications.

The basic protocol allows a customer to sign up for unlimited use of some subscription service for a period of time but prevents the service from determining when he has used the service or what he has accessed. At the same time, mechanisms are provided that make it difficult for the customer to share his subscription with others and leaves him vulnerable to detection if he should do so. Next we introduce a simple protocol for electronic tokens. This protocol can be incorporated into the basic protocol to accommodate subscriptions that include the possibility of pay-per-use transactions.

3.1 Basic Unlinkable Serial Protocol

In this protocol we begin with the assumption that the customer has adequately satisfied the vendor to obtain an account. That is, he has provided the vendor with sufficient proofs of identity, cash, proofs of creditworthiness, etc. We assume that the customer has an associated identifier C for the account, whether or not his identity is actually known by the vendor. (He may in fact have different identifiers for different

accounts.) We assume that he also has at this point an associated signature key for his account, K_C^{-1}. (We use square braces to indicate digital signatures and curly braces to indicate encryption. Thus, '$[X]_K$' refers to data X signed with key K and '$\{X\}_K$' refers to X encrypted with key K. No cryptographic relation between, e.g., K_V in $\{X\}_{K_V}$ and K_V^{-1} in $[X]_{K_V^{-1}}$ is assumed. We use over-lining to indicate blinding: e.g., '$\overline{X}$' refers to the result of blinding X, for use with the appropriate signature key.)

3.1.1 Initially Obtaining Service Certificates

Message 1 $C \rightarrow V$: $[\textit{Request for certificate of type } S,\ C, \overline{h(N_1)}]_{K_C^{-1}}$

Message 2 $V \rightarrow C$: $[\overline{h(N_1)}]_{K_S^{-1}}$

The signature key used here by the vendor is assumed to be used only for signing such certificates. It is also subject to periodic renewal. Service signature keys have a published expiration time. All certificates should be used or exchanged by that time. We will see that there is no need to verify the structure of the blinded hashed nonce. If the client substitutes anything inappropriate the result can only be an invalid certificate.

When the customer wants to make use of the service, he conducts the following protocol with V.

3.1.2 Redeeming Certificates and Obtaining New Ones

Message 1 $C \rightarrow V$: $\{\textit{Request for transaction of type } S,\ [h(N_i)]_{K_S^{-1}}, N_i, K_{CV}\}_{K_V}$

Message 2 $V \rightarrow C$: $\{\textit{Approved}$ OR $\textit{Not approved}$ OR $\textit{Audit}\}_{K_{CV}}$

Message 3 $C \rightarrow V$: $\{\overline{h(N_{i+1})}\}_{K_{CV}}$

Message 4 $V \rightarrow C$: $[\overline{h(N_{i+1})}]_{K_S^{-1}}$

Message 5 $C \rightarrow V$: $\{\textit{Ack } \overline{h(N_{i+1})}\}_{K_{CV}}$

Message 6 $C \leftrightarrow V$: $\{\text{Transaction}\}_{K_{CV}}$

Here K_{CV} is a key that is used to protect the integrity of the session. If the response in message 2 is *Not approved*, e.g., if the certificate was previously used or is not of valid form, then the protocol terminates.

3.1.3 Audit and Not approved

If the response is *Audit*, then a special audit occurs in which C must present some proof of identity within a short period of time. If this is satisfactory, a new certificate is issued. If it is not satisfactory or if C does not comply, then the protocol terminates, and the certificate is logged along with a note that it was used during a failed audit. In either case, no transaction takes place so that audited customers are not linked to specific transaction requests. The main purpose of audits here is to serve as a secondary deterrent to sharing a subscription with a nonsubscriber. (The primary deterrent is the inconvenience of passing the certificate back and forth between those sharing as compared with the cost of obtaining another subscription itself.)

By exercising the audit check frequently or at strategic times, the vendor can learn both the client's usage frequency and patterns. This might allow the vendor to correlate later transactions (and possibly earlier

transactions) with the particular client. The client might counter this limitation by employing a masking scheme on top of the basic protocol. However, this can considerably increase the load on the subscription service. Clients might also counter such vendor analysis by delaying ordinary transaction requests for a random amount of time following an audit. This places no extra burden on the subscription service but may cause customers inconvenience substantially beyond that of audits themselves. Since audits are a secondary deterrent to abuse, they might be conducted infrequently. The tradeoffs between threats to anonymity and the deterrence affect on subscription sharing are difficult to assess a priori. Thus, exactly how frequent to make audits is currently difficult to say.

Notice that if a customer is ever caught during an audit having given away his certificate, he effectively forfeits his subscription. This is because that certificate can never be used again, and no new certificate is issued to continue the subscription. Off-line appeal mechanisms may be available for customers who, for example, lose certificates or secret nonces.

The response to a request for service might be *Not approved* for a number of reasons. These include that the nonce does not match the submitted certificate and that the certificate is not valid for the service requested. Alternatively, the certificate submitted might use an expired key. If the client is a valid subscriber who never received an initial certificate for the current key, this should be reflected in the vendor's records. The client can then get an initial certificate in the usual manner. Off-line appeal will again be necessary for clients who feel they have been refused a legitimate transaction request.

3.1.4 Recovering from Broken Connections

We will consider connection breaks occurring from the end of the protocol to the beginning. If a connection breaks after a new certificate has been acknowledged (message 5), the client can simply initiate a new transaction with the new certificate. If a connection breaks after C receives message 4 but before V receives message 5, the client can again simply initiate a new transaction. (If the connection is not actually broken, but V did not receive message 5, V will refuse to process any request from C (i.e., message 6) until C sends him an acknowledgement.)

Before this point in the protocol the client will not yet have received a new certificate. So, recovering from any connection breaks that occur prior to this point in the protocol will involve replaying the protocol. The vendor should keep a record of each protocol run until he receives the acknowledgement in message 5. When reestablishing a connection C should inform V that this is an attempt to reestablish a broken connection and then replay the protocol exactly, except that each replayed message should contain a field inside the encryption indicating that it is a reconnect message. (Should more than one attempt be made to reconnect, this field must indicate which attempt it is, i.e., second, third, etc.) V should abort the protocol if any of the message fields differ from the previous connection (other than by the reconnect field). If the connection breaks after V has sent an *Audit* in message 2, V should keep a record of the protocol run even if C properly identifies himself upon reestablishing the connection. It may be that a cheater broke the connection and then quickly notified the legitimate client of the audit. The vendor will need to decide what to do, e.g., if several such breaks have been logged for a given C at exactly that point in transactions.

Notice that the customer need never identify himself when a broken connection occurs (unless an audit had already been stipulated by the vendor). Thus, he need not worry about being associated with a given transaction. Even his attempt to reconnect following a connection break in a run where a specific transaction request was made cannot be associated with that request. This is because the actual transaction occurs after a new certificate is issued. And, if a connection were to break after a new certificate was issued, he would initiate a reconnection using the new certificate.

3.2 Service Key Management

For unlinkable protocols to work, it is important that authorization keys are not "closely" associated with clients. For example, we do not want the vendor to be able to uniquely associate a service key with each client, which would enable the vendor to associate transactions with clients.

3.2.1 Committing to Service Keys

A straightforward technique to overcome this vulnerability requires the vendor to publicly commit to all public authorization keys. This can be achieved by publishing information, at regular intervals, at a unique location "well known" to all potential clients of the service. An example publication format for each service consists of the service type, expiration time, and signature confirmation key for signatures associated with this service, together with a one-way hash of these.

3.2.2 Subscription Termination

Other than as a general security precaution, the primary reason to change service keys is to facilitate expiration of subscriptions. When keys expire, our only current mechanism is to have clients obtain new certificates just as they did when signing up for a service initially. Service expiration can be structured in several different ways, each with advantages and disadvantages. We will present some of these and briefly mention some of the tradeoffs. Which is most acceptable will depend on particular aspects of application and context. For the purposes of discussion let us assume that the standard period of subscription is one year divided into months.

3.2.3 Subscription Expiry

One option is to have annualized keys that start each month. In other words, there are twelve valid service keys for the same service at all times. This is convenient for the customer and similar to existing subscription mechanisms; however, it partitions those using a service into twelve groups, reducing the anonymity of customers accordingly. This may or may not be a problem. If subscriptions are annualized to quarters this reduces the threat to anonymity, but this might still be unacceptable. And, it reduces customer flexibility about when subscriptions can begin.

An alternative is to have monthly keys good for all subscribers. Subscribers obtain twelve seed certificates when they subscribe, one for use in each month of the succeeding year. This does not reduce anonymity as the last option did. On the other hand, it requires that customers keep track of the multiple certificates and requires issuing certificates well in advance of their period of eligibility. ¿From the vendor's perspective, the threat of audit becomes much reduced since a cheater will lose at most the current month's certificate. Relatedly, it is that much easier to share a subscription—at least by monthly pieces. Thus, the inconvenience deterrent is reduced slightly as well.

Another option is to have all subscriptions end in the same month. Someone subscribing at other than the beginning of the fiscal year would pay a prorated amount for his subscription. This avoids reductions in anonymity associated with monthly annualized keys. It also avoids the reduced deterrence to cheating associated with monthly keys. But, it reduces customer flexibility in choosing the ending of the subscription. Another disadvantage to this approach is that subscription renewal is now all concentrated at one point in the year, creating extremely unbalanced load on the part of the system handling sign up and renewal. This would probably remain true even if renewing customers were allowed to renew in advance. It could be

diminished by splitting the year in half or even further. This creates the partitioning reduction in anonymity already mentioned.

3.2.4 Early Termination of a Subscription

Terminating a subscription early requires proving that the user is a particular subscriber, and returning a valid certificate. He will not get a new one; so, there is is no way for him to continue using the service. At this point he would presumably be entitled to a refund. Notice that early termination can even be customized, for example, so that it is available only to customer's who have already subscribed for at least a year. (Recall that a customer reveals his identity (or perhaps pseudonym) when he terminates early.) This removes one of the disadvantages of the third option for subscription expiration described above.

We have been describing subscriber termination of a subscription. Vendor termination of a particular subscriber or group is far more difficult. (It may also be less important.) On our current approach the only way to terminate a subscriber is to change the service key(s) for the remainder of his subscription and require everyone else to reinitialize their certificates with the new key. This creates tremendous expense and inconvenience equivalent to what would be necessary if a service key were compromised.

4 Applications of Unlinkable Serial Transactions

Up till now we have been focused on basic subscription services as the application of unlinkable serial transactions. We now explore both expansions of the basic subscription application and other applications as well. As an example, we will set out our first variant on the basic protocol; however, we will simply describe later applications without giving full details on how to adapt the unlinkable serial transactions for them. Generally, it will be straightforward to see how to do so.

4.1 Pay-per-use Within a Subscription

We here describe a means to allow pay-per-use within a subscription. In order to facilitate this application we first describe a simple protocol that allows a vendor to mint his own simple anonymous coins. It can thus be used whether or not a standard e-cash infrastructure is available.

4.1.1 Unlinkable Token Protocol

This protocol produces digital tokens that are to digital cash roughly as tokens in a game arcade are to coins. In fact it is an extreme simplification of some existing digital coin schemes. It might be applicable by itself to an on-line game arcade, jukebox, or movie service. However, our motivation is to use it with the unlinkable serial protocol to enable pay-per-use transactions in a subscription service. We do not here discuss billing for tokens. We assume that the price of tokens was previously agreed. We also assume that since C has an account with V, a request for tokens implicitly authorizes a corresponding charge to C's account.

Message 1 $C \rightarrow V$: $[\textit{Request for tokens, } C, \overline{h(N_1)}, \ldots, \overline{h(N_k)}]_{K_C^{-1}}$

Message 2 $V \rightarrow C$: $[\overline{h(N_1)}]_{K_T^{-1}}, \ldots, [\overline{h(N_k)}]_{K_T^{-1}}$

In this protocol, the tokens are simply hashed nonces. Thus, as in the basic unlinkable serial protocol, the customer need not convince the vendor that the tokens are legitimate by, e.g., unblinding some number of submitted tokens before the vendor will sign the rest. All tokens have the same value, and K_T^{-1} is a key that the vendor uses only to sign tokens. Thus, V need not worry about what he is signing. To spend a token C sends it to V along with the nonce proving that it is his token. The vendor keeps track of spent tokens until they expire. The token signing key should have a published expiration time. All tokens should be used or exchanged by that time to avoid a need to update them.

4.1.2 Unlinkable Serial Protocol for Pay-per-use

Message 1	$C \rightarrow V:$	$\{\textit{Request for transaction of type } S, [h(N_i)]_{K_S^{-1}}, N_i, K_{CV}\}_{K_V}$
Message 2	$V \rightarrow C:$	$\{\textit{Approved}$ OR $\textit{Not approved}$ OR $\textit{Audit}\}_{K_{CV}}$
Message 3	$C \leftrightarrow V:$	$\{\text{Transaction}\}_{K_{CV}}$
Message 4	$C \rightarrow V:$	$\{\textit{Payment}, \overline{h(N_{i+1})}\}_{K_{CV}}$
Message 5	$V \rightarrow C:$	$[\overline{h(N_{i+1})}]_{K_S^{-1}}$
Message 6	$C \rightarrow V:$	$\{\textit{Ack } \overline{h(N_{i+1})}\}_{K_{CV}}$

This is very similar to the basic protocol, except that we move the transaction (message 6 of the basic protocol) to occur before the delivery of a new certificate (messages 3, 4, and 5 of the basic protocol). Also, after the transaction completes and before a new certificate is signed the customer must pay for the transaction. The vendor sends him a billing message as the last message in the transaction. In place of message 4 the customer sends the blinded new certificate together with adequate payment. Payment can use tokens as just described or ordinary anonymous e-cash if that is available and preferred. In this way, the vendor can be guaranteed that payment is received before he signs a new certificate for the customer. However, this opens the possibility that a connection can break during a transaction before a new certificate is received by the customer. The customer still need not worry about revealing his identity to recover from a broken connection, but the vendor will be able to tie the reconnection attempt to the actual requested transaction that occurred during the protocol run in which the connection broke. This may seem a slight inelegance. For an application in which all transactions cost the same amount, transactions can be moved back to the end of the protocol, thus removing the inelegance.

There are alternatives to this protocol. For example, certificates could include a credit balance, which must be periodically paid. Payment would be made as a transaction. There is no harm in this transaction identifying the customer because it is only for payment purposes. The main limitation on this approach is that the credit balance is monotonically increasing. This may allow the vendor to link transactions and even to tie them to particular customers.

4.2 Contracting Out Subscription Management

Vendors may be interested in making available the anonymity afforded by our approach but may be less enthusiastic about the necessary overhead of maintaining a subscription, e.g., keeping track of spent certificates. Along with the ordinary overhead of maintaining subscriptions, handling billing, etc., vendors may choose to hire out the management of subscriptions. It is straightforward to have the vendor simply forward transaction requests to a subscription management service, which then negotiates the business (certificate management) phase of the protocol with the customer. Once this is completed, the transaction phase can proceed between the vendor and the customer as usual.

4.3 Multivendor Packages and Discount Services

For multivendor packages one can purchase what is effectively a book of coupons good at a variety of individual vendors. The way a coupon book would work is that vendors will authorize the package vendor to issue certificates for their services. Customers then engage in a protocol to obtain the basic certificates.

If the coupons in the book are meant to be transferable, there is nothing more to the protocol. If, however, they are not, we must add a serial unlinkable feature to make sharing more cumbersome. In this case, when a customer submits a certificate for a service he must also submit a package certificate. The package certificate must be updated as in the basic protocol. Service certificate are not to be updated: they can only be redeemed once. Vendors could all be authorized with the necessary key to update the package certificate. Alternatively, the processing of the certificates could be handled by the package issuer as in the contracting-out application of unlinkable serial transactions just given. Notice that individual vendors need not be themselves capable of producing even coupons for their own services. It is enough that they can confirm the signatures associated with their services.

Package books such as just described often offer discounts over vendors' basic rates as a sales incentive. Another form of discount is one that is made available to members of some group. Unlinkable serial transactions are useful for allowing someone to demonstrate such membership without revealing his or her identity. Depending on the application, the various vendors offering discounts can sign new certificates or signing can be reserved for some central membership service in association with any request for discount at a vendor. The latter case is again similar to the contracting-out application above.

4.4 Membership and Voting

The example just mentioned shows that the basic idea of unlinkable serial transactions can have application outside of commercial concerns. Specifically it should be useful for any application for which membership in some group must be shown, and where the inconvenience of sharing a serial certificate and the risk of audit outweighs the advantages of spoofing group membership. These might include some applications requiring proof of age or residency.

As another example, consider a voter registration certificate. At voting time, the voter spends his certificate, is issued a new certificate, and votes. The new certificate is signed by a key that becomes valid after the current voting period expires, so voters can't vote twice. In this case, there is no possibility of sharing the certificate for a single election. If there is concern that formerly eligible voters continue to vote once their eligibility has expired, certificate keys could be subject to occasional expiry between elections. Ineligible voters would then be eliminated since they would be unable to register for new seed certificates.

5 Conclusion

In this paper we have presented a protocol that can be used for unlinkable serial transactions. The protocol can be used in several types of commercial services, including unlimited use subscriptions and those incorporating some kind of pay-per-use transaction. Unlinkable serial transactions can also be used for multivendor packages and discount services. And, they can be used for noncommercial applications such as voter registration and proof of group membership. Although individuals are anonymous during each unlinkable serial transaction, they can be challenged to produce identification to prevent various kinds of fraud.

Our approach relies on anonymous communication: there is no sense in using anonymous tokens, pseu-

donyms, etc., if identities are revealed by the communications channel. For Web based commerce, the Anonymizer hides the identity of clients. Onion routing also provides anonymity, but in addition protects against traffic analysis and hides anonymity even if some of the nodes in the anonymity service are compromised.

In this paper we have described means to prevent profiling by vendors. But, profiles may be beneficial to both the customer and vendor, e.g., for marketing purposes. Indeed, services such as Netangels and Firefly are available that build customer profiles for this purpose but promise to protect customer privacy. It might be complicated to incorporate such trusted intermediaries with the protocols we have presented. But, decentralizing may ultimately provide better assurance to customers. Profiles can be collected locally at a user's workstation. This lets individuals control their own profiles. An individual could contact a marketer through an anonymous connection (cf. Section 2.2 and request advertisements suited to his profile. Once he closes the connection the marketer can no longer contact him.

Our approach is based on primitives supporting e-cash but is designed to function in a credit card type commercial infrastructure as well. By manipulating what must be trusted and by whom, as compared with their more common applications, we are also able to greatly simplify the use of such primitives in our protocols.

References

[1] Community ConneXion, Inc., http://www.anonymizer.com.

[2] E. Brickell, P. Gemmell, and D. Kravitz, "Trustee-based Tracing Extensions to Anonymous Cash and the Making of Anonymous Change", Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 457–466, San Francisco, California, 22-24 January 1995.

[3] D. Chaum, "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms", CACM 24, 2, Feb. 1981, pp. 84–88.

[4] D. Chaum, "Security without Identification: Transaction Systems to Make Big Brother Obsolete", CACM 28, 10 (October 1985, pp. 1030–1044.

[5] D. Chaum, A. Fiat, and M. Naor, "Untraceable Electronic Cash", CRYPTO88, pp. 319–327.

[6] http://www.cybercash.com. CyberCash is a trademark of CyberCash, Inc.

[7] L. Cottrell, "Mixmaster and Remailer Attacks", http://obscura.obscura.com/~loki/remailer/remailer-essay.html

[8] A. Fasbender, D. Kesdogan, O. Kubitz. *Variable and Scalable Security: Protection of Location Information in Mobile IP,* 46th IEEE Vehicular Technology Society Conference, Atlanta, March 1996.

[9] M. Franklin and M. Yung, "Towards Provably Secure Efficient Electronic Cash", Columbia University CS Technical Report, TR CUCS-018-92, 1992.

[10] D. Goldschlag, M. Reed, P. Syverson, "Hiding Routing Information", Workshop on Information Hiding, Isaac Newton Institute, Cambridge, UK, May 1996.

[11] C. Gülcü and G. Tsudik, "Mixing Email with *Babel*", 1996 Symposium on Network and Distributed System Security, San Diego, February 1996.

[12] T. Okamoto and K. Ohta, "Universal Electronic Cash", CRYPTO91, pp. 324–337.

[13] A. Pfitzmann, B. Pfitzmann, and M. Waidner. *ISDN-Mixes: Untraceable Communication with Very Small Bandwidth Overhead,* GI/ITG Conference: Communication in Distributed Systems, Mannheim Feb, 1991, Informatik-Fachberichte 267, Springer-Verlag, Heildelberg 1991, pages 451-463.

[14] M. G. Reed, P. F. Syverson, and D. M. Goldschlag, "Proxies for Anonymous Routing", 12th Annual Computer Security Applications Conference, San Diego, CA, December, 1996.

[15] D. Simon, "Anonymous Communication and Anonymous Cash", CRYPTO96, pp. 61–73.

[16] Secure Electronic Transaction (SET) Specification. August 1, 1996.

[17] P. Syverson, D. Goldschlag, and M. Reed. Anonymous Connections and Onion Routing, to appear *Proceedings of the Symposium on Security and Privacy*, Oakland, CA, May 1997.

Unlinkable Serial Transactions

Paul Syverson, NRL
Stuart Stubblebine, AT&T Labs-Research
David Goldschlag, NRL

Objective

Let a person prove that he is in some group, without

- revealing his identity, or
- linking his appearances.

But protect against fraud.

Applications: Subscriptions and Memberships

Flat rate subscriptions:

- **Online information services**
- **Subscriber discounts & premiums**

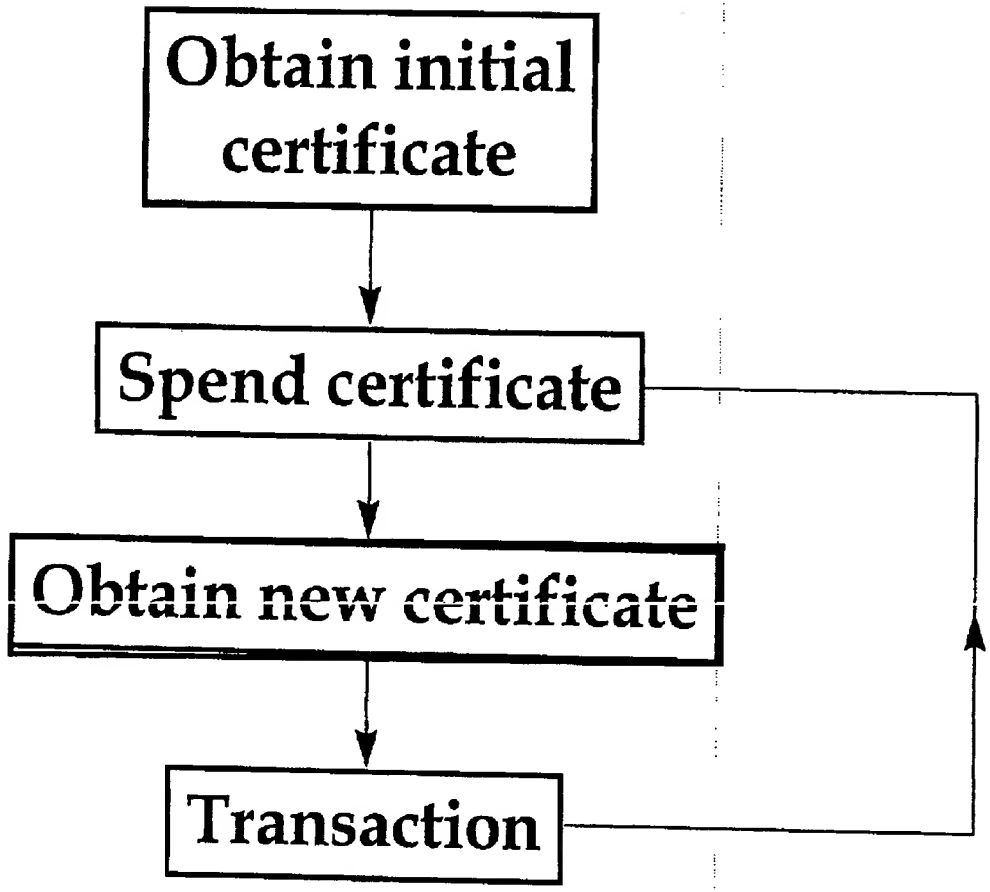
Memberships:

- **Proof of age**
- **State residency**
- **Voter registration**

Summary of Solution

Registration Phase

Transaction Phase



Some Complications

How do we

- discourage sharing subscriptions?
- make subscriptions expire?

Background

Electronic cash techniques:

- Blind signatures
- Signing the hash of a nonce

Transferability is not desirable.

Anonymous communication.

The Protocol: Registration Phase

- A certificate is a signed hashed nonce.
- The signing key determines its value.
- To spend a certificate, show the nonce.

The Protocol: Transaction Phase

Auditing

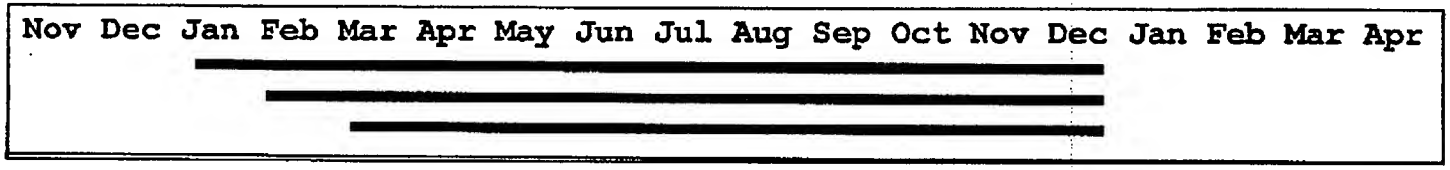
Sharing subscriptions is inconvenient.

Audit may further discourage sharing:

- To obtain a new certificate, the user must prove that he is a subscriber.
- The transaction is always skipped.

Subscription Expiration: Prorating

- All subscriptions end at the same time.
- Prorate late starts.

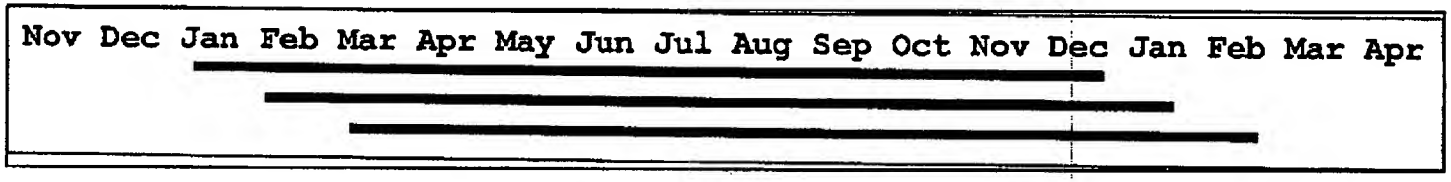


Deficiency:

- Inflexible subscription terms.
- Busy subscription season.

Subscription Expiration: Partitioning

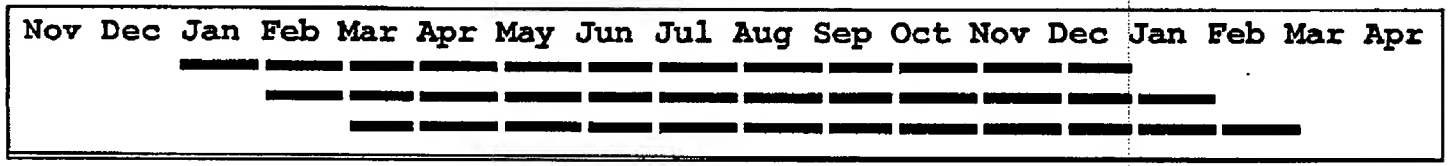
Services use signature keys with overlapping lifespans.



Deficiency: Partitioning subscribers reduces privacy.

Subscription Expiration: Monthly Certificates

- Services use a series of signature keys.
- At registration, issue 12 certificates.



Deficiencies:

- Audit penalty is less severe.
- Short signature key lifetimes may make sharing convenient.

Key Compromise

One solution: Resubscribe everyone.

Key compromise can be efficiently and transparently handled by standard techniques like backup keys.

Administrative termination of a subscription is similar.

Recovering from Broken Connections

Replay *certificate renewal* subphase:

- **Use the same blinded hash and key.**
- **Vendor must store blinded hash until it receives the *Ack*.**

Application: Voting

At voter registration:

- **Obtain initial certificate.**

At polls:

- **Spend a certificate.**
- **Obtain a new certificate that cannot be spent until after the polls close.**

Pay-per-Use Applications

- Purchase tokens.
- Spend during transaction.

This supports services like:

- Lexis/Nexis

Video Rental

Rental transaction:

- Renew certificate.
- Spend tokens for deposit.
- Rent video.

Return transaction:

- Renew certificate.
- Return video.
- Get deposit refund as new tokens.

Forfeiting a deposit ends a subscription, so deposits need not be high.

Entertainment Book

- At registration, issue a personalized coupon book.
- Coupons may be spent during a transaction.

Risk: Coupon book may be shared among subscribers.

Summary

- Transactions are unlinkable.
- Subscribers' privacy is assured.
- Vendor is protected against fraud.

Applications:

- Subscription and memberships

One-time Passwords

At each login, renew the certificate.

Advantages:

- **Passwords are never reused.**
- **The system does not know the password.**

The goal here is not privacy!

Attachment C

UNITED STATES PATENT AND TRADEMARK OFFICE

COMMISSIONER FOR PATENTS
UNITED STATES PATENT AND TRADEMARK OFFICE
WASHINGTON, D.C. 20231
www.uspto.gov

APPLICATION NUMBER	FILING DATE	GRP ART UNIT	FIL FEE REC'D	ATTY.DOCKET.NO	DRAWINGS	TOT CLAIMS	IND CLAIMS
09/635,778	08/11/2000	2161	1596	2685/5681	5	27	13

FILING RECEIPT

OC000000005494267

Kenyon & Kenyon
1500 K Street NW
Suite 700
Washington, DC 20005

Date Mailed: 10/23/2000

Receipt is acknowledged of this nonprovisional Patent Application. It will be considered in its order and you will be notified as to the results of the examination. Be sure to provide the U.S. APPLICATION NUMBER, FILING DATE, NAME OF APPLICANT, and TITLE OF INVENTION when inquiring about this application. Fees transmitted by check or draft are subject to collection. Please verify the accuracy of the data presented on this receipt. **If an error is noted on this Filing Receipt, please write to the Office of Initial Patent Examination's Customer Service Center. Please provide a copy of this Filing Receipt with the changes noted thereon. If you received a "Notice to File Missing Parts" for this application, please submit any corrections to this Filing Receipt with your reply to the Notice. When the PTO processes the reply to the Notice, the PTO will generate another Filing Receipt incorporating the requested corrections (if appropriate).**

Applicant(s)

David M. Goldschlag, Silver Spring, MD ;
Stuart Gerald Stubblebine, Lebanon, NJ ;
Paul F. Syverson, Silver Spring, MD ;

Continuing Data as Claimed by Applicant

THIS APPLICATION IS A CON OF 09/025,802 02/19/1998 PAT 6,108,644

Foreign Applications

If Required, Foreign Filing License Granted 10/21/2000

Title

System and method for voting

Preliminary Class

705

AT&T PATENT
NOV 15 2000
DOCKET
DIRECT
NOT DOCK
PREVIOUSLY
PREVIOUSLY

Data entry by : THOMAS, SHEILA **Team :** OIPE **Date:** 10/23/2000